

マルチメディア絵本制作における 新しい試みの検討 その3

——インラインフレームによる基本枠組み制作——

宮 森 孝 治

1. はじめに

本学幼児教育科1年次後期の専門教科「教育情報処理演習」（幼稚園教諭2種免許必修）において、実際の教育・保育に生かすための演習のひとつとして、ホームページ形式を利用した基本的なマルチメディア絵本の制作を行なっている。

今回、同科1年次・2年次通年専門教科「専門基礎特別演習・専門特別演習（マルチメディア入門）」（履修学生21人）において、「教育情報処理演習」の内容を深めた形で、より効率的に完成度の高いマルチメディア絵本を制作するための方法として、文献[1]および文献[2]で論じたDynamicHTMLやJavaScriptを用いたマルチメディア絵本制作を基本としながら、インラインフレーム構成で絵本の基本枠組みの制作を行なった。

本論は、マルチメディア絵本制作の方法についてのまとめを行ない、基本枠組みを利用した絵本制作の試みの検討を行なった内容である。

2. マルチメディア絵本制作の方法

2-1. 絵本構成の種類

(1). 単純構成の絵本

コンピュータを使用した絵本の中で、最も簡単な方法として、本格的なマルチメディア絵本とは呼べるものではないが、パワーポイントを使う方法やアニメーション合成した画像のみでも絵本として制作可能である。

a). パワーポイントを利用した絵本

プレゼンテーションソフトPowerPointを利用し、それぞれのページに画像を挿入し、

ページ切り替え効果を組み入れたスライドショーで紙芝居的な絵本ができる。

b). アニメーション画像のみの絵本

図1に示すとおり、複数の画像ファイル（GIF形式）を作成し、これらをフリーソフトAnimation GIF Makerを使用して、ひとつのアニメーション合成の画像ファイル（GIF形式）を作り、その画像のみの構成の絵本とする方法である。

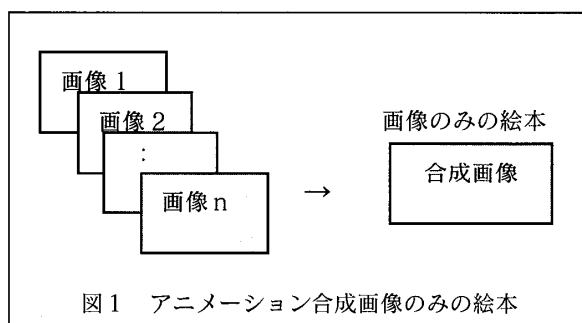


図1 アニメーション合成画像のみの絵本

(2) ホームページ形式の絵本

ホームページ形式を利用した絵本（以下、一般的な呼称であるWeb絵本と記す）の構成である。この方式を用いれば、動く画像や音楽、文字情報等を組み入れることができるので、マルチメディア絵本が制作できる。これには単一頁で構成したものと複数頁で構成したものがある。

a). 単一ページ構成のWeb絵本

①. 合成画像を組み込んだ構成の絵本

単一のページにアニメーション合成した画像を組み入れた構成で、音楽や文字情報等も組み入れることが可能である。（図2に示す）

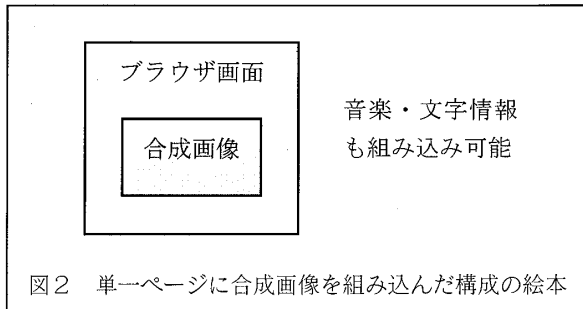


図2 単一ページに合成画像を組み込んだ構成の絵本

②. 複数画像を切り替えた構成の絵本

単一のページ（HTML形式にJavaScript記述を追加）に複数の画像（1.GIF～n.GIF）を一定時間毎に自動的に順次画像が切り替わっていく構成である。また、JavaScript記述を変え、飛び先のページをランダムに変えることができることを利用したゲーム的要素を含んだ絵本の制作も可能である。（図3に示す）

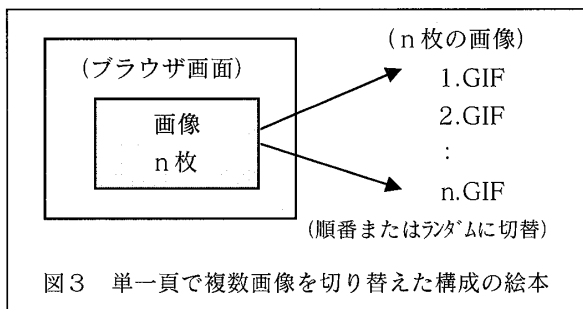


図3 単一頁で複数画像を切り替えた構成の絵本

b) 複数ページで構成したWeb絵本

図4に示すとおり、ページ毎に画像を割り当て、複数枚のページで構成する絵本である。この方式は画像や音楽、文字情報等を各ページ毎に組み入れることができ、完成度の高いマルチメディア絵本が実現できるということで、今回の制作にはこの構成を用いることとした。

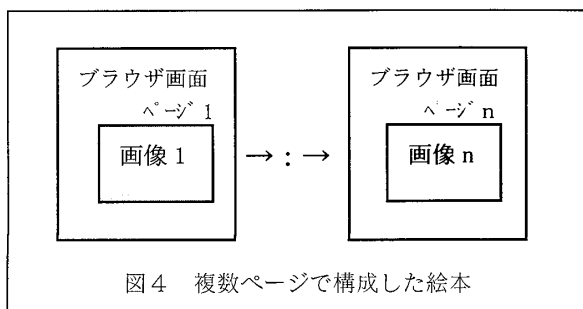


図4 複数ページで構成した絵本

2-2. ホームページ形式絵本の内部構成

(1). 立体的重ね合わせ構成

文献[1]及び[2]の中で詳細に述べているが、図5に示すように、背景、本体1・2、前景といった複数の画像（部品）を平面の位置（XY座標）と、前後のZ方向、すなわち立体的に配置して、ひとつのページを作り上げる。同じ画像を別ページで再利用しながら、効率的に絵本全体を仕上げていくことが可能となる。また画像単位で修正・追加・削除ができ、画像の共通化や分業化が計られ、効率的に作業できることなどの利点がある。今回、ブラウザ画面の左上のポイントを基準に、上端(top)から50ピクセル左端(left)から50ピクセルの位置から配置し、最大の画像（背景）を横480ピクセル、縦360ピクセルの大きさに固定して制作し、その他の画像はその大きさの範囲内でページの適切な位置に配置する構成にした。（詳細は文献[1]参照）

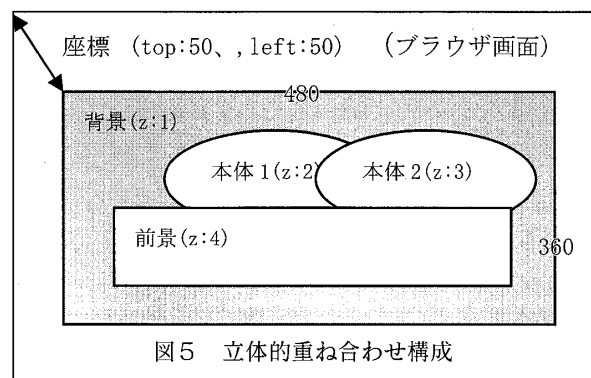


図5 立体的重ね合わせ構成

(2). 画像の動き

画像に動きを加えるには、アニメーション合成画像を使用する方法の他に、<marquee>タグを利用し、パラメータ設定により、左右や上下に移動させる方法と、Dynamic HTMLのダイレクトアニメーション移動（詳細は文献[1]参照）を利用し、パラメータ設定により、任意の位置に移動させる方法がある。

(3). 画像の各種表示効果

それぞれの既存の画像にフィルタを掛けたり、拡大・縮小など様々な効果を組み込むことが可能である。（詳細は文献[1]参照）

(4). マウス操作による画像変化

Dynamic HTMLを記述することによって、マウスの動き（画像の上に載る、クリックなど）

によって、他の画像が表示されたり、説明文を表示したり等、さまざまなインタラクティブ効果が期待できる。(詳細は文献[1]参照)

(5). 音楽やナレーション挿入

サウンド（音楽・ナレーションなど）を組み入れるなど様々な効果を追加しながらマルチメディア絵本として仕上げる事が可能である。これにはMIDやWAV、MP3のファイル形式の音源が利用できる。

(6). ページの切替の自動化と切替効果

〈HEAD〉部分に表示時間とページの飛び先を記述することによって、絵本のページが自動的に切り替わる。今回は、基本枠組みの中に、ページの切り替えが組み入れてあるので、これは利用しない。また、ページの切り替えのタイミングに下から捲れる等といった様々なランジション効果（文献[1]参照）を入れることもできる。

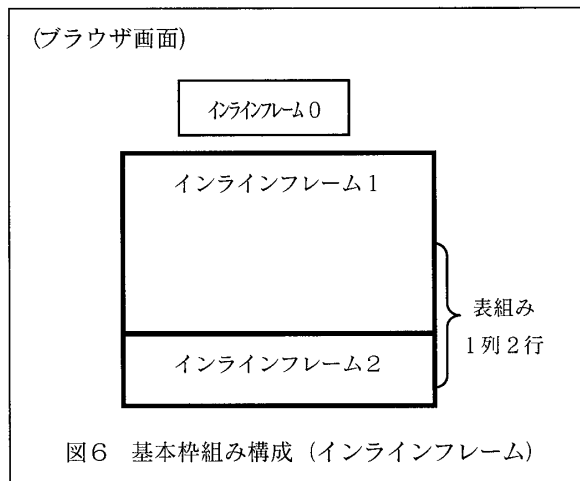
2-3. 絵本の基本枠組み制作

(1). インラインフレーム構成

従来の〈DIV〉タグによる各画像の固定番地で構成される立体的重ね合わせ構成の場合、ディスプレイの解像度やウインドウサイズによって、絵本本体部分を中央に配置することができなかった。複数ページに分割するフレーム構成を用いる方法もあるが、今回の基本枠は、ひとつのページでの構成したほうが分かりやすくなるということで、インラインフレーム構成にすることで検討した。ブラウザの中央に絵本本体が表示されるように、タイトル用、絵本用、文章用に各インラインフレームを中央に配置する。

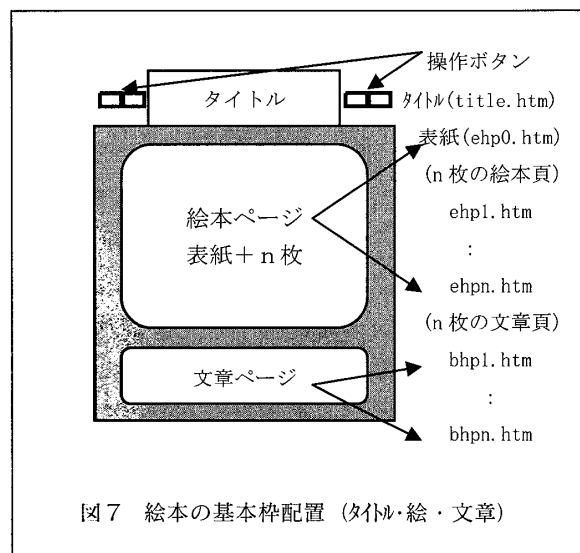
図6に示す構成のように、タイトルをインラインフレーム0に、絵本のページをインラインフレーム1に、文章部分をインラインフレーム2に配置できるよう、HTMLで記述した。

このソース表示は〈記述例1〉に示すように、絵本の基本部分として、インラインフレームで構成しており、JavaScript部分は外部ファイル化し、比較的単純化されている。



(2). 絵本枠制作

インラインフレーム1と2内に枠画像を共通枠として作り、絵本のページと文章のページを紙芝居的な飾りを入れ、全体の絵本としての見栄えも合わせてできるような構成にした。この枠は、各自絵本の内容によって、自由に色や絵で飾れるように、色付けしていない枠を用意した。但し、今回の制作にあたっては、初めての試みということで、なるべく単純な構成で、基本枠組構成のインラインフレーム1の絵本ページに文章を挿入し、インラインフレーム2は隠し、インラインフレーム1のみの構成で制作することとした。(図7に示す)



<記述例1> 基本枠組みページのソース表示

<pre> <html> <head> <title>絵本の基本枠</title> <script src="change.js"> <!--外部ファイル参照--> </script> </head> <body background="back.gif"> <P align=center> <input type="button" name="top" value="TOP" onclick="tophtml()"> <input type="button" name="pre" value="BACK" onclick="backhtml()"> <iframe src="../page0/title.htm" height="38" width="450" marginwidth="0" marginheight="0" scrolling="no" name="frame0" frameborder="1" border="0"></iframe> <input type="button" name="next" value="NEXT" onclick="nexthtml()"> <input type="button" name="auto" value="AUTO" onclick="autohtml()">
 <table border="0"> <tr> <iframe src="../page0/ehp0.htm" height="510" width="580" marginwidth="0" marginheight="0" scrolling="no" name="frame1" frameborder="1" border="2"></iframe> <iframe src="../page0/bhp0.htm" height="0" width="580" marginwidth="0" marginheight="0" scrolling="no" name="frame2" frameborder="1" border="0"></iframe> </tr></table> </center> </body> </html> </pre>	<p><ソース説明></p> <p>←①JavaScriptの外部ファイル宣言</p> <p>←②TOPボタンと操作</p> <p>←③BACKボタンと操作</p> <p>←④タイトル表示のインラインフレーム指定</p> <p>←⑤NEXTボタンと操作</p> <p>←⑥AUTOボタンと操作</p> <p>←⑦絵本表示フレーム</p> <p>←⑧文章表示フレーム</p>
---	--

(3). 操作ボタンの配置

図8に示すように、タイトル表示の左側にTOPボタン（絵本の表紙に戻る）とBACKボタン（1頁前に戻る）を、右側にNEXTボタン（1頁先に進む）とAUTOボタン（表紙から自動で絵本が1頁ずつ捲られていく）を配置した。この結果、制作時の見直しや修正がやりやすくなった。また絵本提示する時、手動切替（NEXTボタン）と自動切替（AUTOボタン）の選択が可能になった。また、この切替ボタンの他、キーボード上の矢印キー「→」（1頁進む）、「←」（1頁戻る）でも操作できるようになっている。

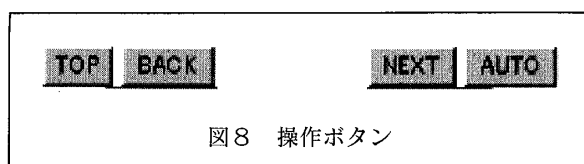


図8 操作ボタン

(4). JavaScript記述の外部ファイル化

初心者にとって、JavaScript記述を完全に理解するのは少し難しい。そこで、ソースの記述をなるべく単純化して理解しやすくすることが必要となる。その方法として、<記述1>の基本枠組みのソースからJavaScript記述部分を外部ファイルchange.jsとし、ブラックボックス化することによって、この部分の概略機能と切替時間の設定や最大ページ数の設定方法を説明するだけで済む構造にした。

外部ファイルの内容は、<記述例2>に示すように、ページの表示切替処理が中心であるが、各種操作ボタンの処理や左右矢印キーの処理も追加している。このようなイベントによって何らかの処理を行なう場合、JavaScriptの記述を用いれば、比較的簡単に作成できる。

<記述例 2> 外部 Java Script ファイル change.js ソース表示	
<pre> var i=0; /*mode counter*/ var a=0; /*auto_mode flag*/ var p=10; /*set max_page counter*/ var t=15000; /*set auto_change_time(msec)*/ /*key_event-sub*/ function checkCSRKey(){ if(event.keyCode==39){nexthtml();} /*right_yajirusi_key*/ if(event.keyCode==37){backhtml();} /*left_yajirusi_key*/ } window.document.onkeydown = checkCSRKey; /*display-sub*/ function chghtml(){ parent.frame1.location.href="./page"+i+"/ehp"+i+".htm"; /*frame1_refresh*/ parent.frame2.location.href="./page"+i+"/bhp"+i+".htm"; /*frame2_refresh*/ } /*next-sub*/ function nexthtml(){ -略- } /*back-sub*/ function backhtml(){ -略- } /*top-sub*/ function tophhtml(){ -略- } /*automode-sub*/ function autohtml(){ -略- } </pre>	<p><ソース説明></p> <p>←①モード(ページ)カウンタ</p> <p>←②AUTO モードフラグ</p> <p>←③最大ページ数設定</p> <p>←④ページ切替時間設定</p> <p>←⑤キーイベントサブルーチン キー「→」:次頁切替処理 キー「←」:前頁切替処理</p> <p>←⑥キーイベント有効</p> <p>←⑦表示処理サブルーチン ・フレーム 1 絵本頁表示 ../pagei/ehpi.htm ・フレーム 2 文章頁表示 ../pagei/bhpi.htm (iは 0～最大ページ)</p> <p>←⑧NEXT ボタン処理</p> <p>←⑨BACK ボタン処理</p> <p>←⑩TOP ボタン処理</p> <p>←⑪AUTO ボタン処理</p>

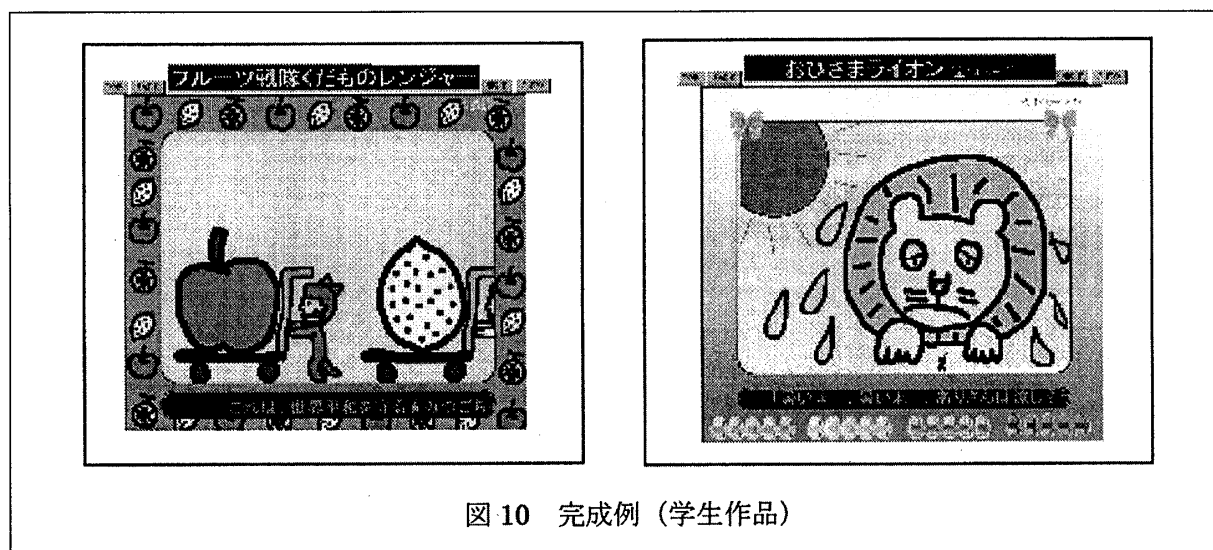
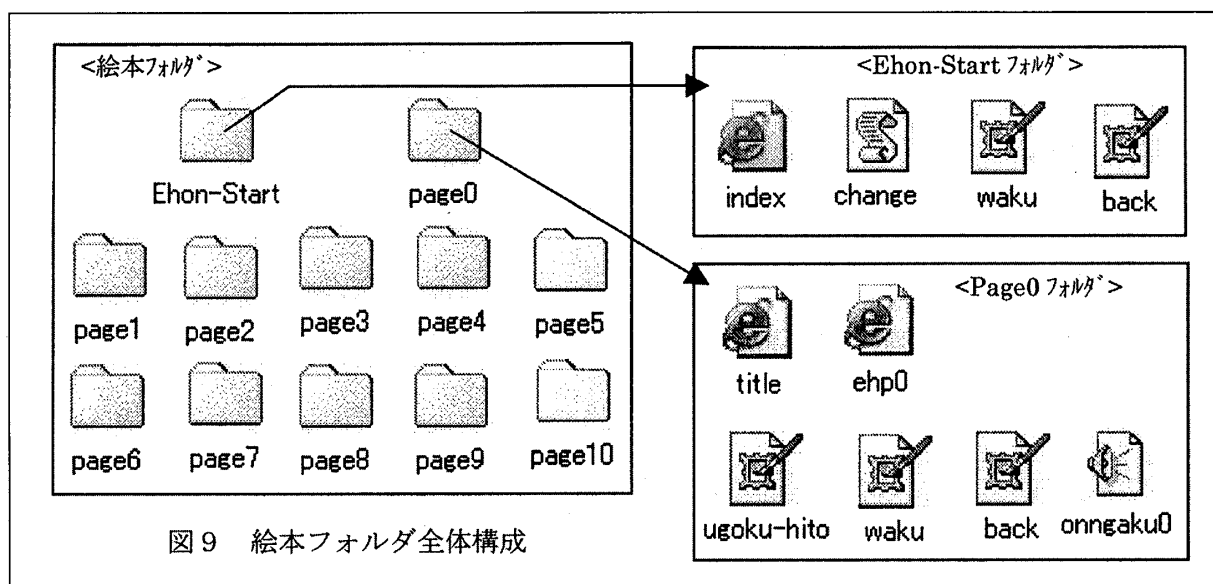
(5). 絵本フォルダの構成

今回の絵本制作は、図 9 に示すような絵本フォルダ構成として、各グループに配布し、これを基に絵本の各ページの制作に取り組んだ。

各グループはEhon-Startフォルダ内において、作品のイメージに合った枠(waku.gif)の画像制作と、最大ページ数と切替時間の設定変更にはJavaScript記述のchange.jsを編集(修正)する。絵本の表紙はpage0のフォルダ内のtitle.htmとehp0.htmのページの制作を行ない、絵本各ページpage1～最大ページ(10ページ程度)それぞれのページの制作を行なった。

(6). 作品完成例

学生の作品の完成例の一部を図10に示す。全体で11グループ(21人)に分かれ、10ページから16ページに及ぶ作品が完成した。童話の内容を一部アレンジしたものもあったが、ほとんど自分達だけで考えたオリジナルの内容が主で、全体的な出来映えとしては、昨年より格段に完成度が高い作品となった。これは今回の方式を使用することにより、外面的な構成にとらわれず、内容を工夫することに専念できたことも要因のひとつではないかと考える。



3. まとめと今後の課題

コンピュータを使った絵本制作には、紙芝居的に画像を単に切り替えていくといった比較的簡単なものから、高価なソフトと高度な技術を駆使して本格的な電子絵本制作まで、様々なものがある。本学において数年前から絵本制作にあたって、コンピュータをあまり得意としていない学生が、一般的なコンピュータ環境の中で、マルチメディア絵本を制作する方法について検討し、実践してきた。

今回は、従来の個々の画像を立体的に重ね合わせる構成を基本にしながら、完成度の高い絵本作りとして、基本枠組みを用いた方法を取り

入れた。制作にあたっては、DynamicHTMLの基本的な知識が必要であるが、JavaScriptについては外部ファイル化したことによって、ブラックボックスとなり、余り気にする必要がなくなり、コンテンツ作りや工夫した画像作りに専念できていた。

基本枠組み構成のねらいとしては、第一に各グループとも全体的な構造（ブラウザのサイズに関わらず中央に絵本が表示）が統一でき、作品としての外観の仕上がり具合が良くなることがあり、第二としては、操作ボタンを設けたことにより、NEXT、BACKボタンによって、各ページの出来上がりやページ間の繋がりが能率的にチェックでき、修正しやすくなったことと、

AUTOボタンによって、作品が自動的に切り替わり、完成度の高い絵本制作が可能になること、また第三として、タイトルや絵本、文章のそれぞれのページを分けそれぞれを個別に制作することができ、分業化や共通化、能率化を目的とした。

実際の制作の取り組みとしては、基本枠組みの構成を把握するのに多少戸惑いがあったようだが、最初から全体構成ができていたので、最大ページ数とページ切替時間の設定だけ行なえば、各ページを個々に仕上げるだけで、全体ができてしまい、ほぼ当初のねらいどおりの結果が得られた。反省点として、2年次の後期から本格的な卒業作品の制作に取り掛かったので、時間的な都合でナレーションの吹き込みができなかった作品もあったが、これを除けば、当初の目的の動く画像、音楽、文字情報など駆使したマルチメディア絵本として、各グループとも完成していた。発表会において、お互いのグループの内容説明や感想、反省点など討議し、それぞれ工夫を重ねながら内容や絵作りで苦労した分、達成感を味わったようだ。特に、ナレーション吹き込みまで組み込んだグループの作品は、今まで以上に完成度が高かった。

これらの作品は、ホームページにて学内外に公開するとともに、本講座の学生には、全作品(CD-ROM)を配布する。これらの作品を専門の職場(幼稚園、保育所等)に就いた時、機会があれば観賞させ、園児等の様子を注意深く観察し、今後とも工夫した作品を各自制作して、教材として活用することを期待する。

今後の取り組みとしては、単純構成の絵本や今回の絵本等を実際の園児等に観賞してもらい、絵本の種類によってどのような反応があるか等について研究を行ないたい。また、この絵本制作の仕組みを、他分野の教材作成等への応用ができないかも検討したい。

<参考文献及び引用文献>

- [1]宮森孝治 マルチメディア絵本制作における新しい試みの検討 (2001) 盛岡大学短期大学部紀要
- [2]宮森孝治 マルチメディア絵本制作における新しい試みの検討(2) -JavaScriptの基礎的活用法- (2003) 盛岡大学短期大学部紀要
- [3]富田憲範 最新HTMLタグマスター (1998) ジャストシステム
- [4]㈱アंक著 改定新版Dynamic HTMLポケットリファレンス (2000) 技術評論社
- [5]古籾一浩 Java Script例文活用辞典 (2001) 技術評論社
- [6]岡蔵龍一／半場方人 詳解HTML&Java Script辞典 (1999) 秀和システム
- [7]丸の内とら 初体験Java Script (2001) 技術評論社
- [8]C & R 研究所 HTML+JavaScript+CSS上級テクニック集 (2002) ナツメ社

<今回使用した主なアプリケーションソフトウェア>

- 1)ブラウザ (Internet Explorer 6.0)
- 2)画像編集 (花子フォトレタッチ ジャストシステム)
- 3)アニメーション合成 (Animation GIF Maker Ver0.65 Freeware Nobuhiro Hattori)
- 4)ソース編集 (メモ帳)